An Exponential Lower Bound on the Complexity of Regularization Paths

Bernd Gärtner
Institute of Theoretical
Computer Science
ETH Zurich, Switzerland
gaertner@inf.ethz.ch

Joachim Giesen
Friedrich-Schiller-Universität
Jena, Germany
giesen@informatik.uni-jena.de

Martin Jaggi
Institute of Theoretical
Computer Science
ETH Zurich, Switzerland
jaggi@inf.ethz.ch

ABSTRACT

For a variety of regularization methods, algorithms computing the entire solution path have been developed recently. Solution path algorithms do not only compute the solution for one particular value of the regularization parameter but the entire path of solutions, making the selection of an optimal parameter much easier. It has been assumed that these piecewise linear solution paths have only linear complexity, i.e. linearly many bends. We prove that for the support vector machine this complexity can indeed be exponential in the number of training points in the worst case.

Keywords

Regularization Paths, Solution Paths, Complexity, Parameterized Optimization, Support Vector Machines

1. INTRODUCTION

Support vector machines (SVM) and related kernel methods have been applied successfully in many optimization, classification and regression tasks in a variety of areas as for example signal processing, statistics, biology, surface reconstruction and data mining.

These regularization methods have in common that they are convex, usually quadratic, optimization problems containing a special parameter in their objective function, called the regularization parameter, representing the tradeoff between small model complexity (regularization term) and good accuracy on the training data (loss term), or in other words the tradeoff between a good generalization performance and overfitting. In particular the C- and ν-SVM versions with both ℓ_1- and ℓ_2-loss [5, 7], support vector regression [23], the LASSO [24], the one class SVM [22], ℓ_1-regularized least squares [15], and compressed sensing [10] are all instances of *parameterized* quadratic programs (pQPs) of the form

$$\begin{array}{lll} \mathbf{QP}(\mu) & \text{minimize}_x & x^TQx + c(\mu)^Tx \\ & \text{subject to} & Ax \geq b(\mu) \\ & & x \geq 0, \end{array} \qquad (1)$$

where $c : \mathbb{R} \to \mathbb{R}^n$ and $b : \mathbb{R} \to \mathbb{R}^m$ are functions that describe how the linear objective function c and the right-hand side b vary with some real parameter μ. Q is an $n \times n$ symmetric positive semidefinite (PSD) matrix, usually the kernel matrix, c is an n-vector (the linear objective function), A is an $m \times n$ matrix (the constraint matrix), and b is an m-vector (the right-hand side).

The task of solving such a problem for all possible values of the parameter μ is called *parametric quadratic programming.* What we want as output is a *solution path*, an explicit function $x^* : \mathbb{R} \to \mathbb{R}^n$ that describes the solution as a function of the parameter μ. It is well known that the solution x^* is piecewise linear in the parameter μ if c and b are linear functions in μ, see for example [19].

Solution path algorithms. An algorithm to compute the entire solution path of the C-SVM has originally been reported by Hastie et al. [14]. [9] gave such an algorithm for the LASSO, and later [17] and [16] proposed solution path algorithms for ν-SVM and one-class SVM respectively. Also Receiver Operating Characteristic (ROC) curves of SVM were recently solved by such methods [3]. Support vector regression (SVR) is interesting as its underlying quadratic program depends on two parameters, a regularization parameter (for which the solution path was tracked by [13, 27, 17]) and a tube-width parameter (for which [25] recently gave a solution path algorithm).

Generic solution algorithms for parametric quadratic programming of the form (1), such as [20, 18] and recently [2], can be applied to the above mentioned applications, instead of using different algorithm descriptions for each variant; Compared to the above mentioned approaches, these generic algorithms also have the advantage that they are able to deal with arbitrary kernel matrices, which do not necessarily have to be invertible.

Complexity of solution paths. Based on empirical observations, Hastie et al. [14] conjecture that the complexity of the solution path of the two class SVM, i.e., the number of bends, is linear in the number of training points. This conjecture was repeatedly stated for the related models in [14, 13, 3, 26, 21, 28, 29, 25]. Here we disprove the conjecture by showing that the complexity in the SVM case can indeed be exponential in the number of training points.

Support Vector Machines. The SVM is a well studied standard tool for classification problems. The primal ν-SVM problem [7] is the following pQP (the related C-SVM and its dual are pQPs of very similar form):

$$\begin{array}{ll} \text{minimize}_{w,\rho,b,\xi} & \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} & y_i(\omega^T x_i + b) \geq \rho - \xi_i \\ & x \geq 0, \\ & \rho \geq 0, \end{array} \qquad (2)$$

where $y_i \in \{\pm 1\}$ is the class label of data point x_i and ν is the regularization parameter. The dual of the the ν-SVM, for $\mu := \frac{2}{n\nu}$, is the following pQP (observe that the regularization parameter moves from the objective function to the constraints):

$$\begin{array}{ll} \text{minimize}_{\alpha} & \sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ \text{subject to} & \sum_{i,\, y_i=+1}\alpha_i = 1 \\ & \sum_{i,\, y_i=-1}\alpha_i = 1 \\ & 0 \leq \alpha_i \leq \mu \end{array} \qquad (3)$$

2. COMPLEXITY OF THE SVM SOLUTION PATH

We will now give an example of a two-class SVM instance, where the solution path has exponential complexity (i.e. exponentially many bends) in the number of training samples, for the case where no kernel is used.

To avoid confusion our example does not just show that some particular algorithm needs exponentially many steps to compute the solution path (as for example the simplex algorithm in linear programming), but indeed shows that any algorithm reporting the solution path will need exponential time, because the path in our example is unique and has exponentially many bends.

Geometric interpretation of the two-class SVM. The dual (3) of the ν-SVM, for $\mu = \frac{2}{\nu n}$ is exactly the polytope distance problem between the reduced convex hulls of the two classes [8], or formally

$$\operatorname{dist}\left(\operatorname{conv}_\mu\left(\{x_i \mid y_i = +1\}\right), \operatorname{conv}_\mu\left(\{x_i \mid y_i = -1\}\right)\right),$$

where

$$\operatorname{conv}_\mu(P) := \left\{ \sum_{p\in P}\alpha_p p \;\middle|\; 0 \leq \alpha_p \leq \mu,\ \sum_{p\in P}\alpha_p = 1 \right\}$$

is the *reduced convex hull* of a set of points, for a given parameter μ, $0 \leq \mu \leq 1$.

We have choosen to present the ν-SVM (instead of the C-SVM), because its regularization parameter ν is straightforward to interpret geometrically as described above. However, this geometric interpretation also holds for the C-SVM as shown by [4], and the correspondence [6] between the two versions implies that our following lower bounds for the solution path complexity do also hold for the C-SVM case.

2.1 A First Example in Two Dimensions

Hastie et al. [14] conjectured that the number of bends in the solution path of a two class SVM is at most $k\min(n_+, n_-)$, where k is some number in the range between 4 and 6 and n_+ and n_- are the sizes of the two classes. First we give an example for an input to the SVM for which the solution path has at least $2(\max(n_+, n_-) - 3)$ many bends, where n_+ and n_- are the sizes of the two point classes. For this, we align a large number n_+ of points of the one class on a circle segment, and align the other class of just two vertices below it, as depicted in Figure 2.

As μ decreases from 1 down to $\frac{1}{2}$, the "left" end of the optimal distance vector, which is a multiple of the optimal $\omega(\mu)$, walks through nearly all of the boundary faces of the blue class. More precisely, the path of the optimal $\omega(\mu)$, for $1 > \mu > \frac{1}{2}$, makes at least twice the number of "inner" blue vertices many bends, which proves the claim.

2.2 The High-Dimensional Case

The Goldfarb cube. It is known that the $2d$ facets of the ordinary unit cube in $\mathbb{R}^d$ can be perturbed slightly such that when we project the resulting polytope onto the last two coordinates, every vertex will be visible in the "shadow". We will denote the two dimensional plane spanned by the last two coordinate vectors by S. This perturbed version of the cube is called the *Goldfarb cube* and already served as an example on which the Simplex algorithm needs an exponential number of steps to find the optimal solution to a linear program [12].

The dual of the Goldfarb cube. The *dual* of a polytope P, or *polar* in terms of Ziegler [30], is defined as $P^* = \left\{y \in \mathbb{R}^d \;\middle|\; x^T y \leq 1 \ \ \forall x \in P\right\}$. In the case that P contains the origin, this is equivalent to

$$P^* = \left\{y \in \mathbb{R}^d \;\middle|\; v^T y \leq 1 \ \ \forall v \in V(P)\right\}$$

and this representation is minimal in the number of constraints. By $V(P)$ we denote the vertices of P, see also [30, Theorem 2.11].

The dual polytope of the cube is the cross-polytope, having linearly many vertices ($2d$ to be precise) and exponentially many facets (2^d of them). The dual of the Goldfarb cube is thus a perturbed version of the cross-polytope, also having only $2d$ vertices but 2^d facets. We initially shift the Goldfarb cube such that the origin lies in its interior.

We now want to translate the "shadow" property of the Goldfarb cube P to its dual polytope D. By looking at the dual constraint $v^T y \leq 1$ for each vertex $v \in V(P)$, it is immediately clear that the dual of the projection of any polytope P (which contains the origin) onto S is exactly the intersection of D, the dual of P, with S. In other words both representations coincide if we restrict ourself just to the two last coordinates. So in our case, the fact that each vertex v_i of the Goldfarb cube, $1 \leq i \leq 2^d$, is visible in the 2-dimensional shadow onto S implies that also the intersection of D with S has exactly that many boundary segments or facets, and the same number of vertices, 2^d each.

Since there are finitely many vertices, and since no vertex of D is lying on any coordinate plane, it is also clear that we can extend the plane S to a thin slab

$$U = \left\{x \in \mathbb{R}^d \;\middle|\; |x_j| \leq \epsilon, \quad 1 \leq j \leq d-2\right\}$$

of thickness 2ϵ, for some $\epsilon > 0$, such that the intersection of D's boundary with this slab still does not contain any

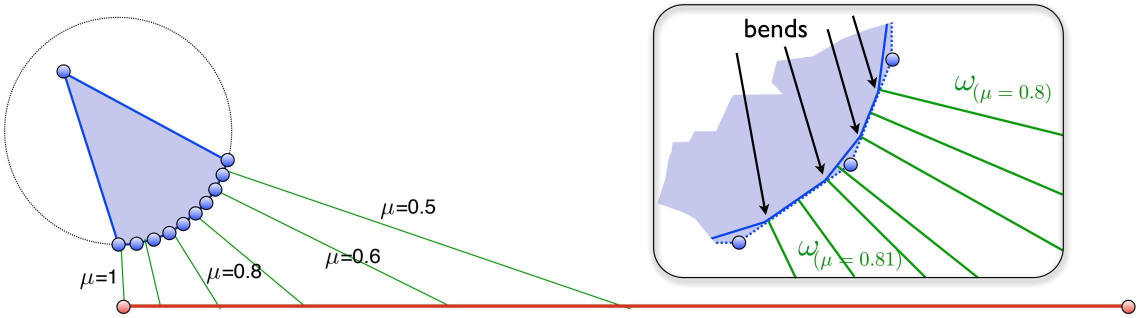

Figure 1: Two dimensional example of an SVM path with at least $\max(n_+, n_-)$ many bends. The green lines indicate the optimal solutions to the polytope distance problem for the indicated parameter value of μ.

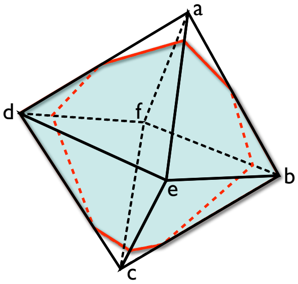

Figure 2: The 3-dimensional cross-polytope D. If you imagine the vertices a and c lying just slightly behind the intersection plane S, and the vertices b and d just slightly in front of the S, then the the plane S intersects all $2^3 = 8$ triangular facets.

vertices. This implies that inside the slab, the combinatorial type of the polytope D is the same as on the plane S, thus any path running around the polytope on its boundary, and staying this slab, runs through the interior of all of the 2^d facets.

Now as we have defined the larger one of our two point classes, to be exactly the described polytope D, which we will use as a replacement of the circle segment in the above 2-dimensional example. Again we let the second polytope Q consist of just a line spanned by two vertices, living in S as in the above example, when we think of S being the 2-dimensional plane which houses Figure 2. By stretching[1]the polytope D away from the plane S, it is easy to achieve that for every point $q \in Q$, the closest point of D to our point q will lie in the slab U.

Having this construction, it again follows directly that when we decrease the regularization parameter μ in the SVM, from 1 down to $\frac{1}{2}$, the solution will pass through at least $\frac{1}{4}$ of the at least 2^d facets of the reduced hull $\text{conv}_\mu(V(D))$ and thus the path will have at least that many bends, which is exponential in $\max(n_+, n_-)$, by our choice of $n_+ = 2d$.

3. EXPERIMENTS

We have implemented the above Goldfarb cube construction using exact arithmetic, and could confirm the theoretical findings. We constructed the "stretched" dual of the Goldfarb cube using Polymake [11], see Figure 3 for a visualization of its intersection with the two dimensional plane S. Having the exact constraint formulation of the polytope, we then used the exact (rational arithmetic) quadratic programming solver of CGAL [1] to calculate the optimal distances for different discrete values of μ. For $d \leq 8$, in all cases we obtained significantly more than $\frac{1}{4}2^{\frac{n_+}{2}}$ bends in the path (we only counted a bend when the set of support vectors strictly changed when going from one μ value to the next).

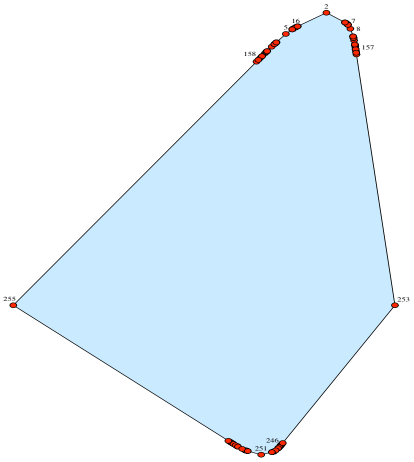

Figure 3: Example for $d = 8$: The perturbed cross polytope of 16 vertices intersected with the two dimensional plane has 256 "bends". Used command sequence in Polymake: `Goldfarb gfarb.poly 8 1/3 1/12; center gcenter.poly gfarb.poly; polarize gpolar.poly gcenter.poly; intersection gint.poly gpolar.poly plane.poly; polymake gint.poly.`

[1] In other words: We scale up all coordinates of the polytope vertices, except the 2 coordinates which define our fixed plane S.

4. REFERENCES

[1] Cgal, computational geometry algorithms library. http://www.cgal.org.

[2] Author. A combinatorial algorithm to compute regularization paths. 2009.

[3] F. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *The Journal of Machine Learning Research*, 7, 2006.

[4] K. Bennett and E. Bredensteiner. Duality and geometry in SVM classifiers. *ICML '00: Proceedings of the 17nd international conference on Machine learning*, 2000.

[5] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, Jun 1998.

[6] C.-C. Chang and C.-J. Lin. Training ν-support vector classifiers: Theory and algorithms. *Neural Computation*, 13:2119–2147, Jan 2001.

[7] P. Chen, C. Lin, and B. Schoelkopf. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

[8] D. J. Crisp and C. J. C. Burges. A geometric interpretation of ν-SVM classifiers. *NIPS '00: Advances in Neural Information Processing Systems 12*, 2000.

[9] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

[10] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586 – 597, 2007.

[11] E. Gawrilow and M. Joswig. Geometric reasoning with Polymake. *arXiv*, math.CO, Jul 2005.

[12] D. Goldfarb. Worst case complexity of the shadow vertex simplex algorithm. *Technical Report*, 1983.

[13] L. Gunter and J. Zhu. Computing the solution path for the regularized support vector regression. *NIPS '05: Advances in Neural Information Processing Systems 18*, 2005.

[14] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research*, 5:1391 – 1415, 2004.

[15] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606 – 617, 2007.

[16] G. Lee and C. Scott. The one class support vector machine solution path. *ICASSP 2007. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2:II–521 – II–524, 2007.

[17] G. Loosli, G. Gasso, and S. Canu. Regularization paths for ν-SVM and ν-SVR. *ISNN, International Symposium on Neural Networks, LECTURE NOTES IN COMPUTER SCIENCE*, 4493:486, 2007.

[18] K. G. Murty. *Linear complementarity, linear and nonlinear programming.* Number Chapter 5. 1988.

[19] K. Ritter. Ein Verfahren zur Lösung parameter-abhängiger, nicht-linearer Maximum-Probleme. *Unternehmensforschung*, 6:149–166, 1962.

[20] K. Ritter. On parametric linear and quadratic programming problems. *Mathematical Programming: Proceedings of the International Congress on Mathematical Programming. Rio de Janeiro, 6-8 April, 1981 / ed.: Richard W. Cottle, Milton Luiz Kelmanson, Bernhard H. Korte*, pages 307–335, 1984.

[21] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Ann. Statist.*, 35(3):1012–1030, 2007.

[22] B. Schölkopf, J. Giesen, and S. Spalinger. Kernel methods for implicit surface modeling. 2004.

[23] A. Smola and B. Schölkopf. A tutorial on support vector regression. *NeuroCOLT2 Technical Report*, (NC2-TR-1998-030), 1998.

[24] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[25] G. Wang. A new solution path algorithm in support vector regression. *IEEE Transactions on Neural Networks*, 2008.

[26] G. Wang, T. Chen, D.-Y. Yeung, and F. Lochovsky. Solution path for semi-supervised classification with manifold regularization. *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 1124 – 1129, Dec 2006.

[27] G. Wang, D. Yeung, and F. Lochovsky. Two-dimensional solution path for support vector regression. *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 993–1000, 2006.

[28] G. Wang, D. Yeung, and F. Lochovsky. The kernel path in kernelized lasso. *International Conference on Artificial Intelligence and Statistics*, 2007.

[29] G. Wang, D.-Y. Yeung, and F. Lochovsky. A kernel path algorithm for support vector machines. *ICML '07: Proceedings of the 24th international conference on Machine learning*, 2007.

[30] G. M. Ziegler. Lectures on Polytopes *Graduate Texts in Mathematics 152, Springer Verlag*, 1995.